

Aplicação de modelos de aprendizado de máquina para classificação de Fake News - Desenvolvimento

Alexandre Augusto Foppa¹, Cristiane Machado¹, Marcelo Reis Bohrer¹

¹Universidade do Vale do Rio dos Sinos (UNISINOS)

alfoppa@hotmail.com, crism.stg@gmail.com, marceloreisbohrer@gmail.com

1. Introdução

Este trabalho consiste no relatório de desenvolvimento da solução descrita na tarefa 01, da disciplina 2020/1 - *Machine Learning*, onde mais detalhes podem ser consultados.

2. Solução proposta

A solução proposta consiste em um modelo treinado para ser utilizado no *Back-end* de uma ou múltiplas aplicações, o qual receberá o texto de uma notícia e retornará um score de 0 a 1, bem como o rótulo mais provável (*'Real'* ou *'Fake'*) - como uma API (application programming interface), podendo ser integrado com sites, aplicativos, etc.

Dada a natureza textual do dado, utilizamos um algoritmo de Processamento de Linguagem Natural (PLN) para extração de features do texto da biblioteca Scikit-learn. Utilizamos o texto sem stopwords, portanto é necessário que o texto consumido pela aplicação tenha, e um momento anterior a predição, suas stopwords removidas. A arquitetura e o método não foram determinados, por não termos um estrutura/modelo de negócios definidos. Além disso, foram removidos 50 registros que não possuíam texto disponível.

Com as features extraídas, testamos diferentes modelos para avaliar o que melhor atende a demanda, sendo eles Decision Tree, Random Forest, Naive Bayes, Extra Trees e Logistic Regression, também através da biblioteca citada. Além disso, também uma Artificial Neural Networks (ANN), utilizando Keras e Tensorflow para avaliarmos o desempenho de um algoritmo de *deep learning*. Os algoritmos citados foram testados com diferentes parâmetros, sendo os resultados mais promissores detalhados na seção seguinte.

2. Aplicação dos algoritmos

Para a aplicação dos algoritmos, foram testadas diferentes partições entre treino e testes, bem como no limite de features extraídas dos textos. Para fins de comparação, disponibilizamos os códigos em Jupyter Notebooks, adotando como padrão uma divisão de 80% para treino e 20% para testes, com 150000 max features. Experimentamos com os parâmetros fornecidos pela documentação conforme o algoritmo (ex.: aumentando e diminuindo número de árvores em random forest, mudando tipos de kernel utilizados, etc), sendo disponibilizado na entrega a versão que teve a melhor performance, com a

única exceção de *Random Forest*, que teve a melhor performance com a redução do max features para 1500.

No teste utilizando *Artificial Neural Networks*, além dos parâmetros, testamos a inclusão e remoção de camadas ocultas. O desempenho dos algoritmos baseados em árvore mostrou uma eficácia significativamente melhor, portanto dedicamos mais tempo a teste de seus parâmetros. Para a entrega final, foram disponibilizadas os códigos de treino e os modelos com maior acurácia obtida no tempo hábil da disciplina, salvos e carregados com a biblioteca pickle.

Além disso, conforme literatura, verificamos uma dificuldade de obtenção de uma boa especificidade - acertar que uma notícia verdadeira é verdadeira se mostrou mais fácil do que identificar uma notícia falsa. Conseguimos um ganho de especificidade mudando o limiar de determinação de *Real* ou *Fake* para 0,425 nos modelos de maior eficácia (*Extra Trees* e *Random Forest*), às custas de eficácia, conforme veremos na seção seguinte.

3. Análises de Performance

Abaixo na Figura 1, temos o comparativo dos resultados entre os modelos aplicados em busca da melhor solução para o problema proposto. Maiores detalhes disponíveis na planilha de métricas anexa.

Métricas por Modelo

Modelo	Acurácia	Especificidade	Sensibilidade	Precisão	F1
Extra Trees	0,8268	0,5766	0,9524	0,8176	0,8799
Extra Trees (limiar 0,425)	0,8098	0,7445	0,8425	0,8679	0,8550
Radom Forest	0,8049	0,5328	0,9414	0,8006	0,8653
Radom Forest (limiar 0,42..	0,7829	0,6934	0,8278	0,8433	0,8355
Logistic Regression	0,7659	0,6496	0,8242	0,8242	0,8242
Decision Tree	0,7610	0,6934	0,7949	0,8378	0,8158
Artificial Neural Networks	0,6927	0,7153	0,6813	0,8267	0,7470
Naive Bayes	0,6854	0,5223	0,7866	0,7263	0,7552

Acurácia



0,6854 0,8268

Figura 1 - Métricas por Modelo

Seguindo com os comparativos, os gráficos abaixo foram gerados no intuito de melhorar a visualização dos resultados obtidos entre os modelos testados em questão.

Acurácia

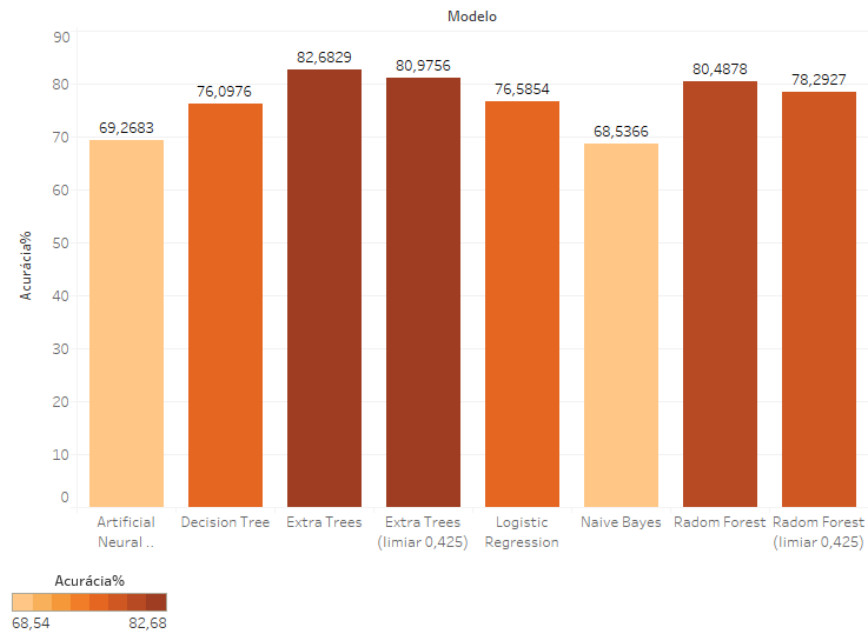


Figura 2 - Acurácia

Especificidade

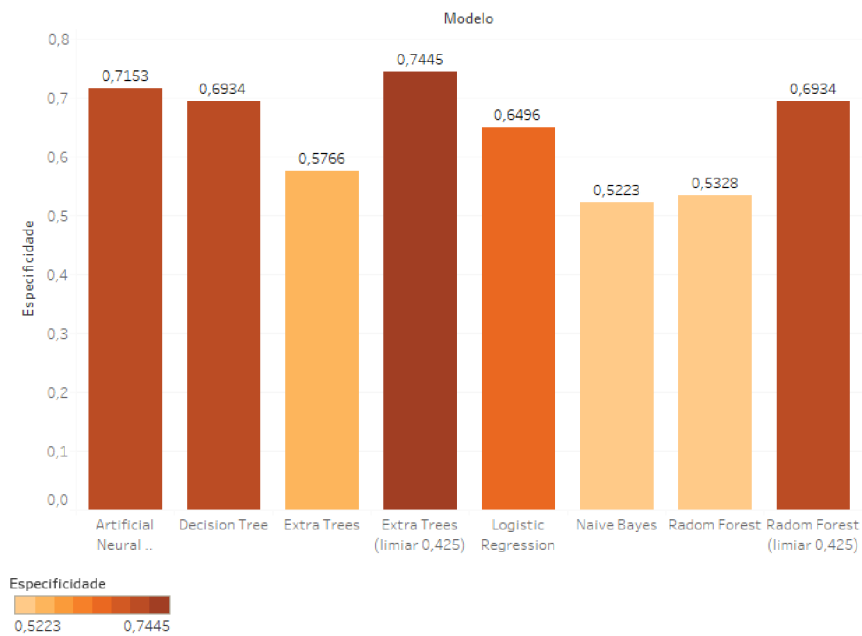


Figura 3 - Especificidade

Sensibilidade

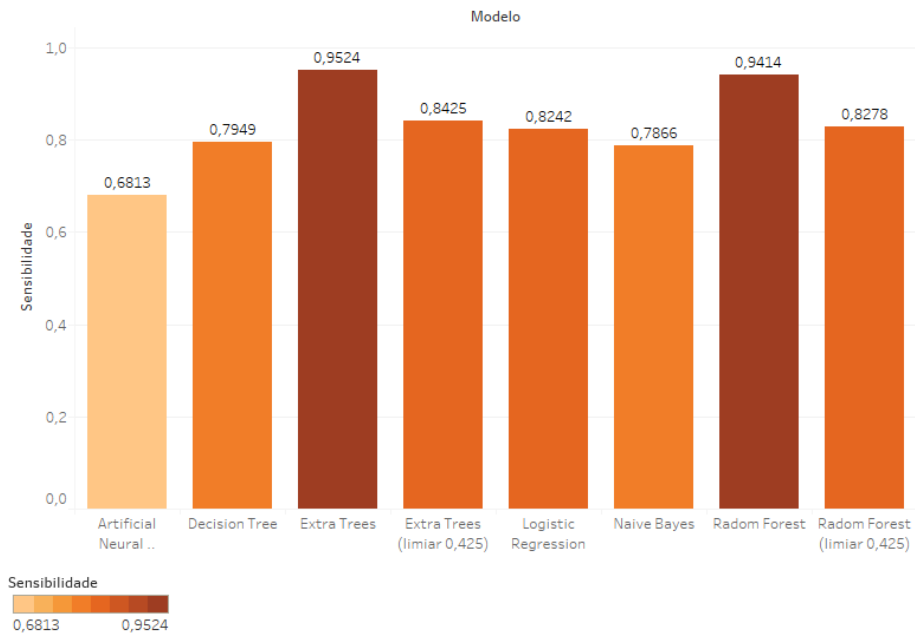


Figura 4 - Sensibilidade

Precisão

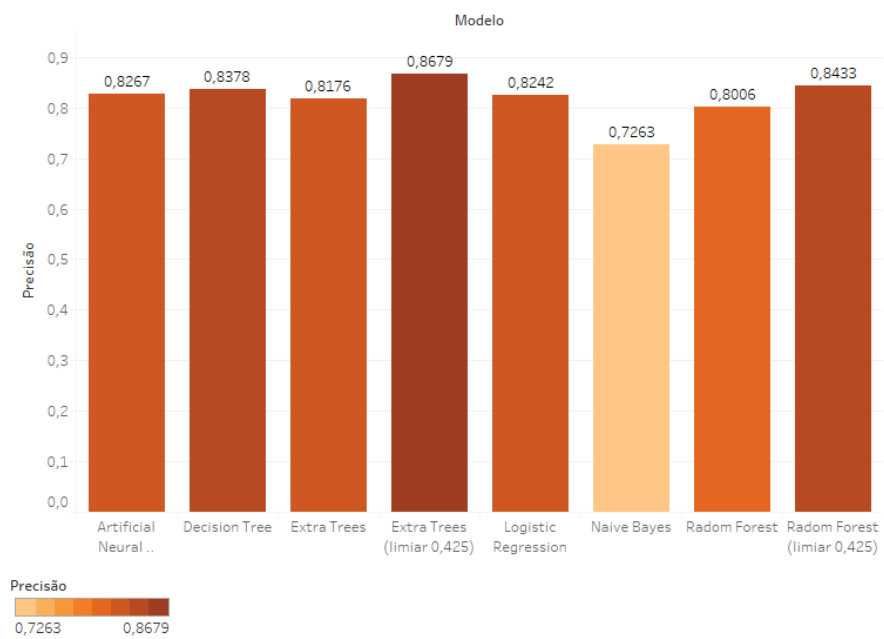


Figura 5 - Precisão

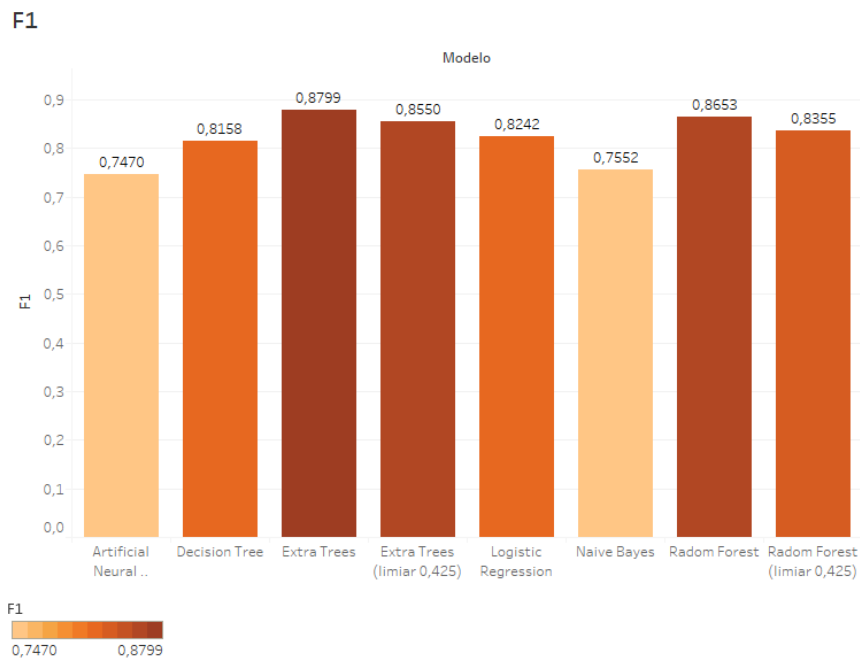


Figura 6 - F1

4. Conclusão

Conforme os resultados do desenvolvimento aplicação corroboram a segunda hipótese de nosso projeto, de que o algoritmo tem uma eficácia aceitável para identificar notícias reais, mas com dificuldade de identificar notícias falsas. Mesmo assim, ele demonstra aplicabilidade para combater fake news, podendo oferecer dos serviços interessantes:

- Utilizar o modelo Extra Trees para identificar com 82,68% de eficácia notícias verdadeiras e as propagá-las, e limitar o alcance das demais até que uma maior análise seja feita, ajudando a balancear quantitativamente a velocidade de disseminação entre notícias falsas e verdadeiras;
- Utilizar o modelo Extra Trees com limiar de 0,425, com maior especificidade, para emitir uma mensagem ao usuário quando a notícia é identificada como falsa pelo modelo.

Ainda é possível concluir que o algoritmo pode se beneficiar de um aumento na quantidade de dados disponibilizados para o treinamento, bem como uma análise com mais tempo sobre as diferentes possibilidades de combinação de parâmetro.

5. Referências

- Avaaz. (2020) “O Brasil está sofrendo de infodemia coronavírus”, https://secure.avaaz.org/campaign/po/brasil_infodemia_coronavirus/, October.
- Edell, Aaron.(2018) “*I trained fake news detection AI with >95% accuracy, and almost went crazy*”, <https://towardsdatascience.com/i-trained-fake-news-detection-ai-with-95-accuracy-and-almost-went-crazy-d10589aa57c>, October.
- Kaggle. <https://www.kaggle.com/>, October.
- Shu, Kay, et. al.(2017) “*Fake News detection on Social Media: A Data mining perspective*”, <https://doi.org/10.1145/3137597.3137600>, New York, ACM, Volume 19, October.
- Scikit-learn.(2020) <https://scikit-learn.org/stable/modules/classes.html>, October.